



Fair Work
Commission

Modern Awards Pay Database

API Integration

Best Practices Guide



Contents

Contents	2
About the Modern Awards Pay Database API	3
What is the Modern Awards Pay Database API	3
What technology standards does the Modern Awards Pay Database API use.....	3
Preparing for Implementation	4
Developer Portal registration for large teams	4
Inactive accounts.....	11
Terms of Use	11
Design, development, and pre-production readiness	11
Design Considerations.....	11
Using the Modern Awards Pay Database API	14
Implementing the Modern Awards Pay Database Webhooks.....	15
Pre-production testing	21
API Support & Governance	22
API support arrangements.....	22
API usage reporting.....	22
API governance and prior version support	23



Fair Work
Commission



About the Modern Awards Pay Database API

What is the Modern Awards Pay Database API

The Modern Award Pay Database (MAPD) API provides users (such as software developers who register with our API) with the ability to call on and retrieve the latest updates to modern award rates within the Modern Awards Pay Database in a structured digital format. The Modern Awards Pay Database API includes access to the following information from the [Modern Awards Pay Database](#):

- Minimum rates of pay for each classification,
- Most penalty rates, overtime, and casual rates,
- Some junior and apprentice rates, and
- Some wage-related and expense-related allowances.

What technology standards does the Modern Awards Pay Database API use

The Modern Awards Pay Database API has been developed in accordance with the Australian Government's API Standard, which leverages modern web API technologies, including:

- HTTP using REST / ReSTful patterns,
- Support for JSON and XML data payloads,
- Transport Layer Security (TLS) version 2 or higher,
- Open API Specification Version 3 (OAS2 and WADL also supported), and
- Webhooks.

As the APIs only provide read-only, publicly available data, we (the Fair Work Commission) have decided to limit authentication and authorisation requirements to API keys. No support is required for OAuth2 or OIDC.



Fair Work
Commission



Preparing for Implementation

Developer Portal registration for large teams

The steps to register on the developer portal are:

- Step 1.** In your web browser, navigate to:
- <https://developer.fwc.gov.au> (Production), or
 - <https://uatdeveloper.fwc.gov.au> (Test)

- Step 2.** Click the “Log in /register” button on the home page.

Fair Work Commission APIs

The Fair Work Commission is Australia's workplace tribunal.

We create awards, approve enterprise agreements and help resolve issues at work.

Modern Awards API's

Access our Modern Awards API to support your payroll and business processes.

Log in / Register


Need help with your login or registration? Log a ticket using our [Contact us](#) form.

[Log In / Register](#)

- Step 3.** This will trigger a pop-up window containing the FWC's Sign-in Window:

Sign up or sign in - Work - Microsoft Edge

https://fwcdevbtcad.b2clogin.com/fwcdevbtcad.onmicrosoft.c...

 Fair Work Commission

Sign in

Sign in with your email address

[Forgot your password?](#)

[Sign in](#)

Don't have an account? [Sign up now](#)



Fair Work
Commission

Step 4. Click the “Sign up now” link, located underneath the blue “Sign in” button:

Don't have an account? [Sign up now](#)

Step 5. Enter your email address and click the blue “Send verification code” button. This will trigger a verification email containing a unique one-time code to be sent to the nominated email address. Ensure to review the Privacy Notice available on this page.

User Details

For information on the collection, use and disclosure of personal information, view the [Privacy Notice](#).

Step 6. Locate the email that will be sent from the msonlineservicesteam@microsoftonline.com email address. Copy the unique code:

Verify your email address

Thanks for verifying your [\[Your email address will be here\]](#) account!

Your code is: 358699

Sincerely,
FWC Dev B2C tenant

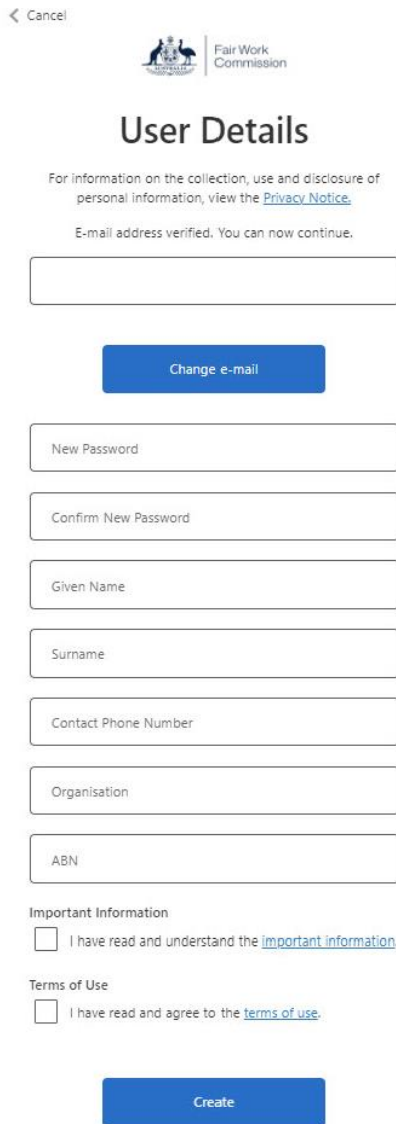


Fair Work
Commission


Step 7. Return to the Sign-Up pop-up window and enter the verification code in the text field provided and click the “Verify code” button:

The screenshot shows a 'User Details' verification screen from the Fair Work Commission. At the top left is the Fair Work Commission logo. The title 'User Details' is centered. Below the title, there is a link to the 'Privacy Notice'. A message states that a verification code has been sent to the user's inbox. There are two input fields: one for the email address (containing the placeholder text '[Your email address goes here]') and one for the 'Verification Code'. At the bottom, there are two blue buttons: 'Verify code' and 'Send new code'.

Step 8. Complete the basic user information:



< Cancel

 Fair Work
Commission

User Details

For information on the collection, use and disclosure of personal information, view the [Privacy Notice](#).

E-mail address verified. You can now continue.

[Change e-mail](#)

Important Information
 I have read and understand the [important information](#).

Terms of Use
 I have read and agree to the [terms of use](#).

[Create](#)

Password Guidance:

The Password you select must adhere to the following rules:

- Between 8 and 64 characters in length
- Contain at least 3 of the following:
 - A lowercase letter
 - An uppercase letter
 - A digit
 - A symbol

Contact Phone Number:

Users are encouraged to include their own phone number for UAT subscriptions. For Production accounts, organisations are expected to provide the contact phone number of an individual or team responsible for the production support of the API client product or service.

Organisation and ABN:

This information is collected to assist us to identify the users of the Modern Awards Pay Database API. This information will be verified, and incorrect information may result in rejection of the sign-up application or cancellation of user accounts.

Terms of Use and Important Information

These important legal documents contain critical information on the use of the Modern Awards Pay Database API. Please read these thoroughly prior to registering. Clicking the blue links will open the documents in a new browser window to enable you to share them easily within your organisation.



Fair Work
Commission

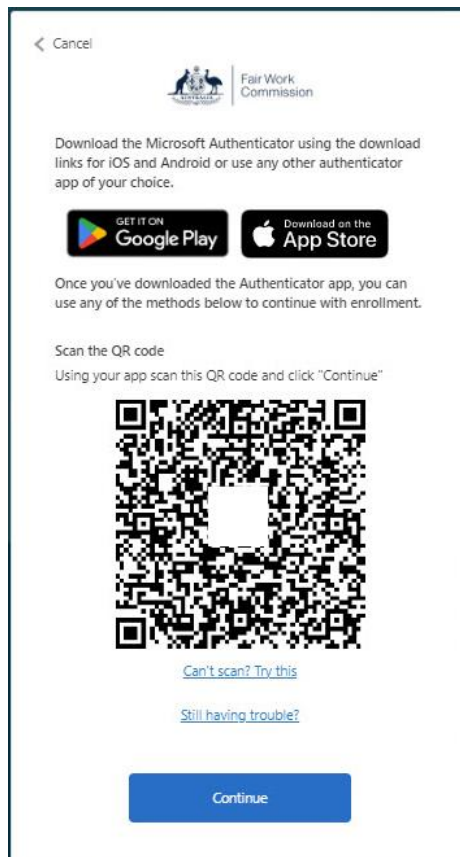
Step 9. Read the [Important information](#); Check the 'I have read the Important Information' statement.

Step 10. Review the [Terms of Use](#). After reviewing them and if you agree to them, check the 'I agree to the Terms of Use' box. You will not be able to use the API if you do not agree to the [Terms of Use](#). It is critical that users subscribing to the Production Modern Awards Pay Database API are appropriately authorised to accept these [Terms of Use](#) on behalf of their organisation.

Step 11. Click 'Create' to register your account. When the registration has been successfully completed you will be prompted to configure multi-factor authentication for your account.

You will see the following screen, open your preferred authentication application, if you don't have a preferred application then you can use the provided links to obtain the free Microsoft Authenticator App from the Google Play or Apple App Store.

Open the App, click to add a new account and scan the provided QR Code.






Fair Work
Commission

Step 12. Complete the steps to add the account to your app. When finished, return to the above screen, and click the “Continue” button.

Step 13. You will be prompted to add the verification code generated by the App. Enter the code and click the “Verify” button.

< Cancel

 Fair Work
Commission

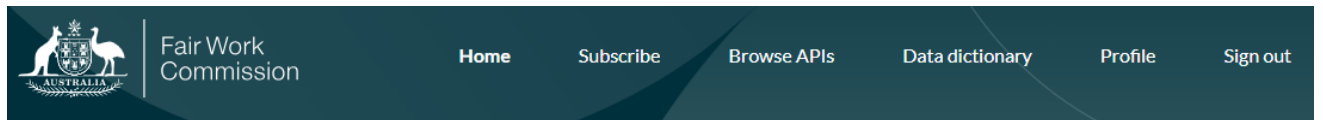
Enter the verification code from your authenticator app.

Enter the 6-digit verification code generated by the the Authenticator app in the box

Enter your code

Verify

Step 14. You will now be returned to the FWC’s Developer Portal and automatically logged in. You will notice the top navigation bar has been updated to include several new items:



Step 15. Click on the “Subscribe” link to sign up a new API client product and obtain your API Subscription Keys.

Step 16. Type your product subscription name and click the “Subscribe” button.

Your subscriptions

You don't have subscriptions yet.

testapi

Subscribe



Step 17. You will automatically be redirected to the 'User Profile' page that will show your new subscription including your Subscription keys, which your will manage from this screen going forward.

User profile

Account details

Email
First name
Last name
Registration date

01/20/2023

Subscriptions

Subscription details	Product	State	Action
Name: testapi Started on: 01/20/2023 Primary key: xx Secondary key: xx	FWC Modern Awards API	Active	Rename Show Regenerate Show Regenerate Cancel

When your subscription is approved, you will also receive a confirmation email which includes some useful information and links.

Step 18. You can click "Show" to display the Primary or Secondary Subscription Keys in clear text.

Step 19. You can now navigate to the API's Link in the Navigation Bar to explore the Modern Awards Pay Database API. When you click on the "Try It" button with the API Explorer if will be pre-populated with your Subscription Keys.

End of Registration Process



Fair Work
Commission

Inactive accounts

We may from time-to-time review the user accounts in the non-production environment and choose to suspend or disable accounts that have been inactive for a period greater than 12 months.

Our [Terms of Use](#) note that the circumstances in which we may terminate user access, including without notice, and the terms on which we provide access to the MAPD API.

Terms of Use

The use of the Modern Awards Pay Database API is governed by the [Terms of Use](#). All Modern Awards Pay Database API users, including production and non-production users, will be required to accept the Terms of Use prior to gaining access to the 'Try it' functionality or directly calling an API endpoint.

It is critical that users subscribing to the Production Modern Awards Pay Database API are appropriately authorised to accept these [Terms of Use](#) on behalf of their organisation.

Design, development, and pre-production readiness

Design Considerations

Understanding the nature of modern awards information

Modern awards cover many working people in Australia and different awards apply to different industries and occupations. Each modern award sets out the minimum pay rates and conditions of employment for those covered by a particular award, including:

- Penalty rates, overtime, and casual rates,
- Annual and other types of leave,
- Hours of work, and
- Allowances.

The Modern Awards Pay Database API provides access to a sub-set of that information in a digital format, particularly providing:

- Minimum rates of pay for each classification,
- Most penalty rates, overtime, and casual rates,



Fair Work
Commission

- Some junior and apprentice rates, and
- Some wage-related and expense-related allowances.

Modern award minimum rates and allowances are subject to an annual review and we may decide to update the minimum rates and resulting allowances during this annual wage review. We may also make other updates at other times throughout the year as other decisions are made by us which update or change those rates.

We have guidance available about what the MAPD API includes and doesn't, why this is important to know and where to get assistance in accessing modern awards. See [Modern Awards Pay Database API](#).

Design for resilience

Whilst we have designed the Modern Awards Pay Database API to be highly available we do not provide an availability commitment to API subscribers. This is outlined in its Terms of Use.

You should design your solutions to be fault tolerant and capable of operating effectively if the Modern Awards Pay Database API is unavailable for a period. When designing your products, you should consider implementing a policy to locally cache Modern Awards Pay Database data for a suitable period of time.

If this functionality is used, we **strongly recommend** you utilise the Webhooks capability to allow you to systematically subscribe to update events and assist you to ensure that your software is remaining up to date with adjustments and changes that may be made to MAPD rates. Further information about the Webhooks capability is provided below. Responsibility for ensuring correct rates for end users, rests with the user.

Design to reduce load on the Modern Awards Pay Database API

Given the range of rates available in the full suite of Modern Awards Pay Database API there is the potential for a large transactional load for stable data, with data changes that are generally infrequent.

To reduce load we have implemented caching mechanisms that respond to most transactions at the front door, and throttling capabilities to minimise load on back-end infrastructure and provide consistent performance for all users.

API users are also required to be responsible users. So, what does that look like?

1. Analyse the business use case that requires Modern Awards Pay Database API data, including modelling:



Fair Work
Commission

- a. the cumulative frequency of use,
 - b. the distribution of requests across modern awards,
 - c. peak transaction periods
2. Implement an appropriate caching mechanism for Modern Awards Pay Database API data, we suggest at least to 24 hours for high frequency API calls.
 3. Leverage cache optimised queries where possible.
 4. Subscribe to Modern Awards Pay Database Webhooks to be notified of data changes. The following subsection will outline good practices for consuming Modern Awards Pay Database webhooks.
 5. Engage early with us to help us understand your expected transaction load and ensure that we are jointly prepared to support a positive experience for Australian employers, employees, payroll practitioners and business advisors. Contact our support team via the [Contact us form](#), please select Modern Awards Pay Database (MAPD).
 6. Understand the importance of your obligations under the [Terms of Use](#) in using and accessing the Modern Awards Pay Database API. This includes but is not limited to awareness that data from the MAPD API must be read in conjunction with the applicable modern award. We have guidance available about what the MAPD API includes and doesn't, why this is important to know and where to get assistance in accessing modern awards. See [Modern Awards Pay Database API](#).

For reference, we throttle requests on a per User/API Key basis:

- 1000 requests per second for HTTP GET calls.
- 500 requests per second for PUT and PATCH calls to webhook endpoints.

Additionally, we have configured the API gateway with throttling policies that leverage IP Address and Regional throttling strategies. If your solution is expected to exceed the above thresholds or traffic is expected to originate outside of Australia then please advise us as soon as practical, prior to the production release of your Modern Awards Pay Database API functionality.

If you exceed the API throttling limit the API will return a “HTTP 429 Too Many Requests” error.



Using the Modern Awards Pay Database API

API Subscription Key

The Modern Awards Pay Database API is a read-only API (apart from Webhooks subscription management) and as such we have made the decision to only implement API Key based authorisation in preference to other more complex methods.

To obtain an API Subscription Key, refer to steps 15 through 19 in the Developer Portal Registration instructions above.

Once an API Subscription Key has been obtained it is the responsibility of the User and their organisation to securely store and manage these API Keys. If an API Subscription Key has been publicly exposed, please notify us as soon as possible and regenerate the Subscription Key by logging onto the Developer Portal, navigating to the Profile page and clicking the “Regenerate” link corresponding to the exposed key.

To use the API key in you solution you MUST include the API Subscription Key value in the following header of all API Calls:

Header Name:	Ocp-Apim-Subscription-Key
Header Value:	[API Subscription Key value]

Understanding errors

When the Modern Awards Pay Database API is unable to process a request and returns an error, the high-level reason for the error will be indicated by the HTTP Response Code. The Modern Awards Pay Database API uses the following response codes:

Code	Meaning
200	OK
201	Accepted
400	Bad Request
404	Not Found



500	Internal Error
-----	----------------

All MAPD API endpoints also support a common error response object, which is aligned with the Australia Government's National API Standard:

Attribute	Data Type	Comments
Errors	Array	-
→ Id	String	A unique identifier for the error event
→ detail	String	A human-readable explanation specific to this occurrence of the problem.
→ Code	String	A textual application-specific error code, for example, "HandshakeFailed"
→ source	Object	An object containing references to the source of the error, optionally including any of the following members: Pointer or Parameter
→ parameter	String	JSON Pointer [RFC6901] to the associated entity in the request document [e.g. "/data" for a primary data object, or "/data/attributes/title" for a specific attribute].
→ pointer	String	A string indicating which URI query parameter caused the error.

Implementing the Modern Awards Pay Database Webhooks

What are webhooks?

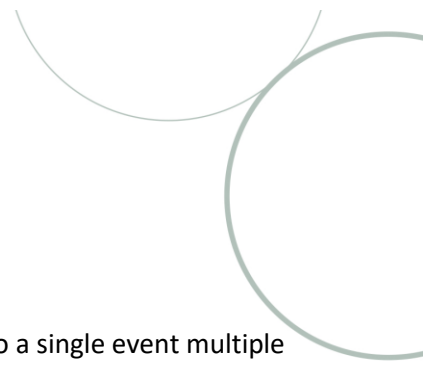
If you are new to webhooks the easiest way to describe a webhook is a subscription to an API resource that notifies your predefined endpoint when that resource has been modified.



Webhooks are still an emerging capability and there is no formalised or agreed international standard for their implementation. As a result, we review industry practices, and techniques in use across the accounting and payroll software sector, leveraging common design features to establish the webhooks capability for the Modern Awards Pay Database API.

The Modern Awards Pay Database Webhooks support users to subscribe to 17 different triggers, designed to provide the API user with maximum flexibility to configure the subscriptions that best suit their business needs.

1. **AllAwards** - All awards.
2. **SingleAward** - Single award specified by `award_fixed_id`.
3. **AllClassifications** - All classifications.
4. **ClassificationsForAward** - Classification related to an award specified by `award_fixed_id`.
5. **SpecificClassification** - Classification identified by `classification_fixed_id` (passed as `entity_fixed_id`).
6. **AllExpenseAllowances** - All expense-related allowances.
7. **ExpenseAllowancesForAward** – Expense-related allowance related to an award specified by `award_fixed_id`.
8. **SpecificExpenseAllowance** – Expense-related allowance identified by `expense_allowance_fixed_id` (passed as `entity_fixed_id`).
9. **AllPenalties** - All penalties.
10. **PenaltiesForAward** – Penalty related to an award specified by `award_fixed_id`.
11. **SpecificPenalty** - Penalty identified by `penalty_fixed_id` (passed as `entity_fixed_id`).
12. **AllWageAllowances** - All wage-related allowances.
13. **WageAllowancesForAward** – Wage-related allowances related to an award specified by `award_fixed_id`.
14. **SpecificWageAllowance** – Wage-related allowances identified by `wage_allowance_fixed_id` (passed as `entity_fixed_id`).
15. **AllPayRates** - All pay rates.
16. **PayRatesForAward** - Pay rates related to an award specified by `award_fixed_id`.
17. **PayRateForSpecificClassification** - Pay rates identified by `classification_fixed_id` (passed as `entity_fixed_id`).



Each subscription is treated independently, meaning that an API user can subscribe to a single event multiple time, specifying a different notification endpoint for each subscription.

The process to subscribe to a webhook

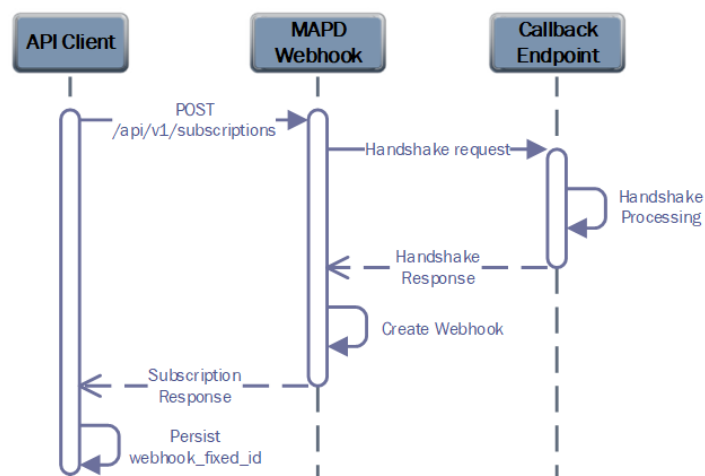
We provide a series of API endpoints to manage your webhooks subscriptions. To subscribe to a webhook you need to have the following pre-requisites completed:

1. Have a callback endpoint configured and publicly available over the internet.
2. Have enabled TLS 1.2 or higher on the callback endpoint.
3. Have configured the callback endpoint to process the Modern Awards Pay Database Webhook Handshake request.

Once the pre-requisites are in place, you will send a HTTP POST request to the <https://api.fwc.gov.au/api/v1/subscriptions> endpoint with the correctly formatted input payload. If successful you will receive a HTTP 201 response, if unsuccessful you will receive a HTTP 400 response code, including an error description in the response payload.

How does the Modern Awards Pay Database Webhook Handshake request work?

We have implemented a Webhook Handshake on signup. This is to prevent someone maliciously signing up a 3rd party endpoint and using the Webhook to generate a Denial-of-Service attack on that 3rd party.



When you submit your HTTP POST request to subscribe to a webhook you will provide the following key information in the payload:



Key	Required	Description
http_verb	Yes	The Modern Awards Pay Database webhook only supports HTTP POST. If you specify another method (e.g., GET, PUT) then the webhook subscribe method will return a HTTP 400 error.
callback_url	Yes	The Modern Awards Pay Database webhook will validate this value. If empty, it will return a HTTP 400 error with the BadRequest message. If the callback_url is invalid (doesn't exist or respond as expected) then the error code will be HandshakeFailed.
hook_on	Yes	The value of this field must match one of the 17 possible webhook triggers.
entity_fixed_id	No	The identifier of the target award sub-entity (rate, allowance, penalty etc).
award_fixed_id	No	The identifier of the target award to which you are subscribing.
webhook_headers (Array)	No	A maximum of five (5) webhook_header objects are allowed.
webhook_headers→ name	Optional	You can provide a header that will be sent with all calls to your endpoint.
webhook_headers→ value	Optional	You can provide a static header value that will be sent with all calls to your endpoint.

The MAPD subscription endpoint will process the POST request and then initiate its own POST request (the Handshake request) to the specified callback_url. The handshake request will contain the following payload structure:



Fair Work
Commission

```
{
  "correlation_id": "0ca96cd3-cef7-4746-8cb9-a1a70d5ef5c1", // unique identifier for this handshake
  "webhook_fixed_id": "480a070d-e3df-4a88-9aec-8389de3e7955", // this will be the fixed_id for your webhook
  "message_type": "HANDSHAKE",
  "payload": null
}
```

Your call-back endpoint will then be expected to process this payload. If your endpoint encounters the `"message_type": "HANDSHAKE"` value, then it is expected to return a HTTP 201 response with a payload containing the `webhook_fixed_id` key and value, that was received. If the Modern Awards Pay Database Webhook cannot negotiate an acceptable TLS version and algorithm set, it will return a HTTP 400 error.

You should also persist the `webhook_fixed_id` and the `callback_secret` value as you will need this in future to maintain your webhook registration. The `callback_secret` can be used in future to validate the authenticity of requests to your callback endpoint.

What happens if my callback endpoint is down when a webhook trigger is sent?

If a webhook is triggered for a resource to which you have subscribed, we will attempt to send a trigger payload to your callback endpoint. If your registered callback endpoint is unavailable or times out, we will perform the following processing steps:

1. We will reattempt the request four (4) additional times, with a short delay between requests.
2. If we are unable to successfully connect (your endpoint does not provide a valid HTTP 200 response) we will mark your webhook subscription as inactive.

While your webhook subscription is marked as inactive, any new triggers will not result in notification to your endpoint. You will need to make a HTTP PUT request to the `/subscription/{webhook_fixed_id}` endpoint to re-enable your webhook. It is your responsibility to ensure that your webhook endpoints are active. We recommend you implement monitoring of your webhook endpoint.



Fair Work
Commission



Recommended practices for Webhooks

Use webhooks to monitor for changes rather than polling the Modern Awards Pay Database API

The Modern Awards Pay Database API has been designed to support short-term peaks in API transaction volumes but is not anticipating that every software product will call the API for every employee, for every pay run. As the Modern Awards Pay Database Data has a relatively low rate of change outside of the Annual Wage Review, we recommend that API users design their products to locally cache high-volume API responses (e.g., base rates, penalties, and allowances etc.) and only call the API to retrieve the latest changes.

As such, we recommend subscribing the relevant Webhook for these items to receive a callback with a few minutes, if the underlying resource changes. This prevents unnecessary polling of the API, which will return the same value on most attempts. Of course, you are welcome to call the API outside of a webhook trigger but use common sense to ensure that you are abiding by the Terms of Use (see [Terms of Use](#)) and acting as a responsible user.

Only subscribe to what matters

The Modern Awards Pay Database API provides the flexibility to subscribe to webhooks at multiple levels and to specific modern awards or classifications. It is recommended that your solution subscribe only to the resources that matter to your solution and product.

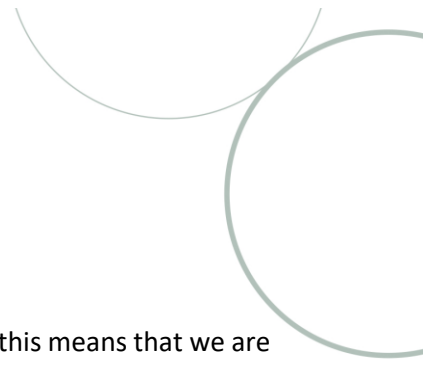
For example, out of the 155 modern awards, 121 are industry and occupational modern awards, the remainder are modern enterprise awards and State reference public sector modern awards. Products targeted to a particular industry or type of employment may not need to reference all awards.

Build a solution to filter and queue notifications, then process

We have built a solution that allows you to set a small set of custom headers and we provide a `callback_secret` when you subscribe to a webhook. You can use these values to filter requests that haven't come from us, dropping them at the door to protect from end-point spamming. Additionally, you may want to use the custom headers to support flags that support you to prioritise updates for certain subscriptions or awards.



Fair Work
Commission



Ideally, we expect your endpoint to respond within two seconds or less. In real time, this means that we are anticipating that you will receive the webhook callback, validate its authenticity, queue it for asynchronous processing by a downstream process, and then send us success response.

Why does the Modern Awards Pay Database API encourage you to queue the request rather than call on API endpoints to download the latest resource while initially processing the webhooks?

That answer is simple, efficient use of resources. We expect to be sending hundreds of thousands of webhook callbacks to different software products in a very short period of time during annual wage review periods. If we must wait for software products to undertake what complex processing logic before we get confirmation that the callback has been delivered, we will have a significant backlog of requests on our outbound queue. This might delay someone else's or your next notification, slowing the overall ecosystem down and delaying the time until your product is up to date.

Managing maintenance windows and outages

You can mark your callback endpoint as offline for maintenance and mark it back online afterwards. This puts you in control and prevents the Modern Awards Pay Database Webhooks from attempting to send you messages and marking your endpoint offline without your knowledge.

However, as with most webhook solutions you might miss something during this period. The best recovery mechanism is to run a query to check what data has changed during this period before you mark your webhooks back online.

Each of our API endpoints currently supports the ability to obtain all current records for a given resource and sort by `last_modified_datetime` in descending order.

Pre-production testing

Are there security requirements or other requirements that need to be demonstrated in an integration?

It is your responsibility to ensure that your product is fit for purpose and secure.

We encourage all software developer organisations using the Modern Awards Pay Database API to actively engage with us as part of launch planning. We will not endorse any specific products or services, but can work



Fair Work
Commission

with you to discuss technical support, for example your anticipated transaction volumes. We can be contacted in relation to the application of particular modern awards and other related matters. See [Modern Awards Pay Database API](#).

It also should be noted that we offer the Modern Awards Pay Database API in accordance with the Terms of Use that you must accept to obtain a UAT and Production User Account and API Subscription Key. The terms of use note we, the Commission can revoke access to the Modern Awards Pay Database API.

API Support & Governance

API support arrangements

The Modern Awards Pay Database API is supported by us on a best effort basis, in accordance with our Terms of Use.

You can check the current operational status of the Modern Awards Pay Database API here:

- <https://api.fwc.gov.au/status>

If you experience issues with the Modern Awards Pay Database API or have a technical enquiry, please contact us by using the [Contact us form](#) (select Modern Awards Pay Database – technical enquiry):

Our support business hours are Monday to Friday 9am to 5pm AEST. We will respond to your enquiry as soon as practicable.

API usage reporting

The FWC's Developer Portal provides statistical usage reports for your API subscription covering an historical period of up to 90 days. (<https://developer.fwc.gov.au/reports>)

These reports enable you to view the following statistics relating to your individual product subscription:

- Request volumes (Success, failure, blocked, total) per subscription,
- Average response time in milliseconds per subscription,
- Request volumes (Success, failure, blocked, total) per operation,
- Average response time in milliseconds per operation.



Fair Work
Commission

API governance and prior version support

We anticipate due to the nature of modern awards that most future enhancements to the Modern Awards Pay Database API will be backwards compatible.

However, it has committed to an N-1 support model where the current version of the API plus the version immediately prior will be supported. API users using older version of the Modern Awards Pay Database API may find they have inconsistent experiences.